Hangout: Sprint #4 Presentation

Antonio | Brandon | Eric | Ashley | Dean

Intro

No updates to Business Requirements Document

Management plan updates: Sprint board, project tracking matrix, burndown chart, sprint retrospective
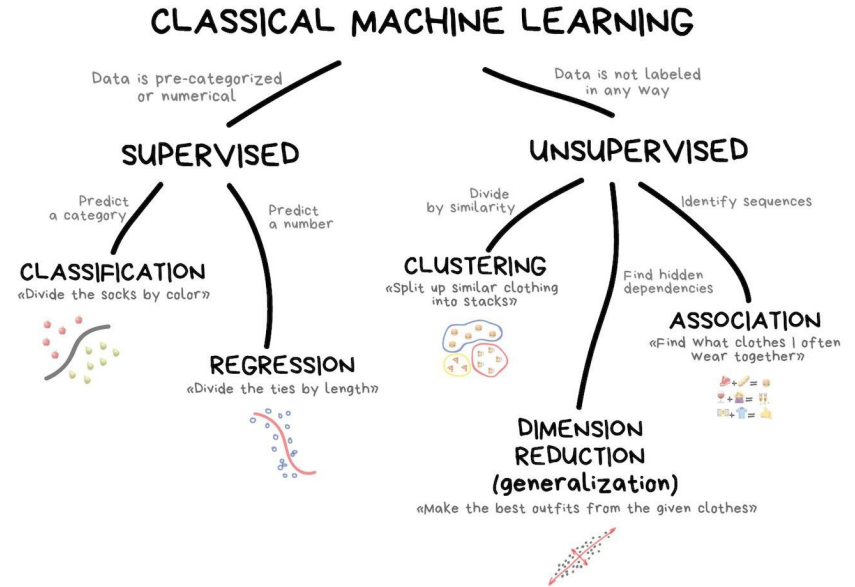
User stories released in sprint:

As a user, I want to be able to create an account.
As a user, I want to be able to sign in to my account.

A decision tree.

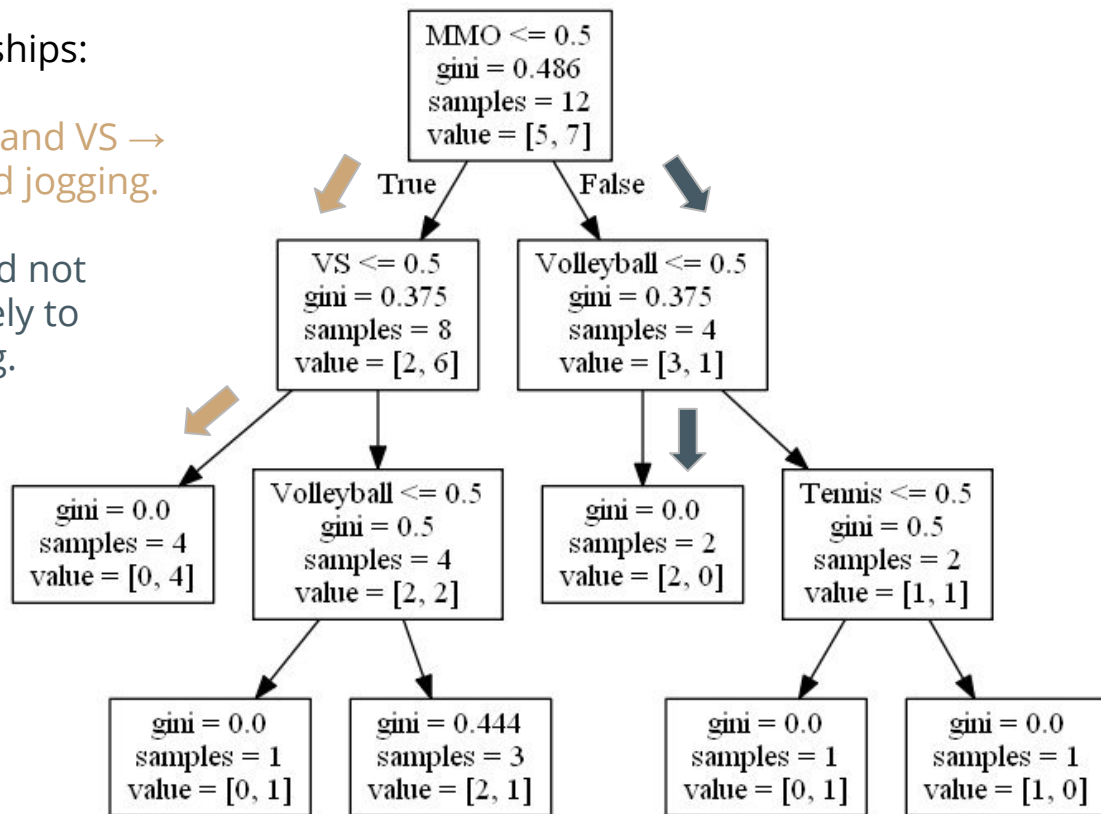The current training (and testing) build examines six features.

```
MMO,VS,Volleyball
Tennis,Swimming,Jogging
```


CLASSICAL MACHINE LEARNING

Data is pre-categorized or numerical

SUPERVISED

Predict a category

Predict a number

CLASSIFICATION
«Divide the socks by color»

REGRESSION
«Divide the ties by length»

Data is not labeled in any way

UNSUPERVISED

Divide by similarity

Identify sequences

CLUSTERING
«Split up similar clothing into stacks»

Find hidden dependencies

ASSOCIATION
«Find what clothes I often wear together»

DIMENSION REDUCTION
(generalization)
«Make the best outfits from the given clothes»

Illustrated Relationships:

0 interest in MMOs and VS → likely to recommend jogging.

Interest in MMO and not volleyball → not likely to recommend jogging.

MMO <= 0.5
gini = 0.486
samples = 12
value = [5, 7]

True   False

VS <= 0.5
gini = 0.375
samples = 8
value = [2, 6]

Volleyball <= 0.5
gini = 0.375
samples = 4
value = [3, 1]

gini = 0.0
samples = 4
value = [0, 4]

Volleyball <= 0.5
gini = 0.5
samples = 4
value = [2, 2]

gini = 0.0
samples = 2
value = [2, 0]

Tennis <= 0.5
gini = 0.5
samples = 2
value = [1, 1]

gini = 0.0
samples = 1
value = [0, 1]

gini = 0.444
samples = 3
value = [2, 1]

gini = 0.0
samples = 1
value = [0, 1]

gini = 0.0
samples = 1
value = [1, 0]

Current tree plot. The target label is jogging.

Source of Training Data

The team plans to generate the data.

Technical Approach:

- Some informal, real instances from people and social media.
- Some statistics (e.g. an interest in swimming correlates with an interest in water polo).

The team plans to deploy as a web API using the Flask framework. It is likely to be written in Python.

Alternatively, the team considered writing the model straight onto the main server.

The team chose an API:

1. It is more clean and kept for the code. The model is meant to support the product, and any ML issues can be resolved separately.
2. If some issue were to arise, the model can be translated and implemented into the code as a resort.

Features

How many total user features are there?

- There are 2 completed features.

What % of the user features do you find appealing or essential to the product?

- 100% of the features are found to be essential to the product. Without these features, Hangout would not function therefore rendering it useless for the user.

Which new feature/user story is the most impressive? If there are multiple such items, please name all of them.

- Being able to Sign In to an account is the most impressive because without being able to create user accounts, Hangout would be far less functional.

User Features

-Hangout is finally starting to launch with real features!

-Primary feature added in this sprint is the hosting, allowing for users to create accounts and sign in on the website, as opposed to injecting the data using Postman.

What is to come?

- Hangout is going to be experiencing upgrades Soon™ in both features and in appearance.
- Because of split focus, development of app features has been slower than intended. Despite this, Hangout remains optimistic that we will deliver a quality product.

Architecture

No updates made to the Architecture/Design Document

With the backend API finally deployed, focus can be turned to finishing user features.

- Utilizes MVC Pattern
- Front end pages currently set up to test whether back end is correctly fetching data
- Plans to use Flux for the Events.js

Code Review

Postman used to test node.js API

- Since API lacks GUI, Postman was used to test fetch requests before front end was completed
    - (GET, POST)

All code submitted into repository was first reviewed via a Pull Request

- API code ordered in MVC

Code Demo

Sprint Goal

Our goal this sprint is to have a functioning product with registration and login that is hosted on a server for users to access.
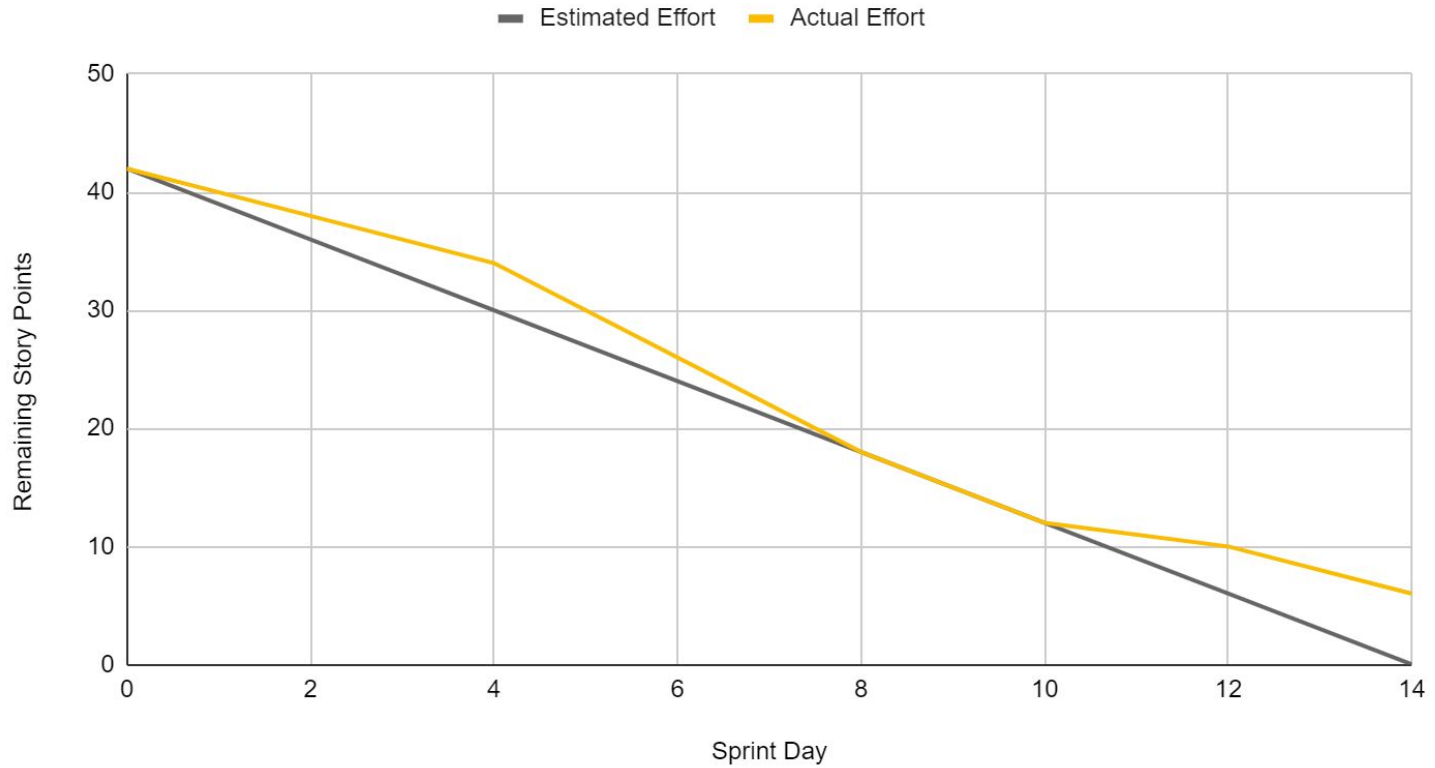
47 User Story Points Planned

41 User Story Points Achieved

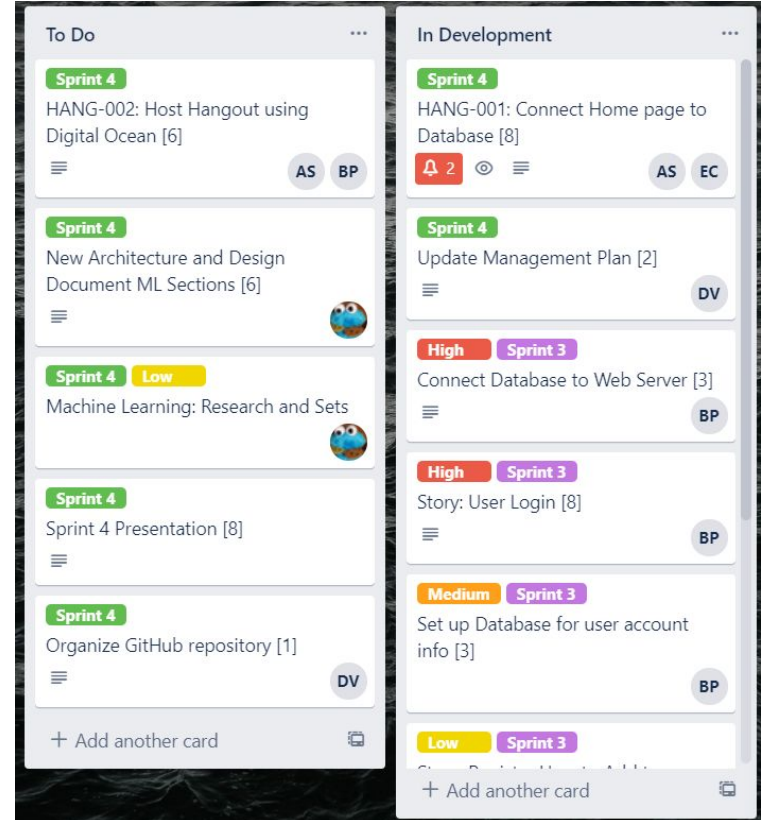User stories categorized: Must have, Should, Could, Won't. All stories implemented this Sprint are "Must have".

Approximate # of "In-Progress": 5

Sprint #4 Burndown Chart

Sprint Retrospective

- This sprint we managed to not only accomplish most of what we set out to accomplish on sprint 4, but also clean up remaining sprint 3 tasks

Sprint Retrospective

What worked well?
    -Many tasks were completed
    -Individual accomplishments were high

What could have been better?

    - A focus on webapp development, not machine learning
    -Slow ramp up time into work